Docket No.: 22058-00003-US
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Bruce Dickens

Patent No.: 5,806,063

Group Art Unit: 2177

Issued: September 8, 1998

Examiner: Jean Homere

For: DATE FORMATTING AND SORTING FOR DATES SPANNING THE TURN OF THE CENTURY

REQUEST FOR REEXAMINATION

Commissioner for Patents
Washington, DC 20231

February 7, 2003

Dear Sir:

This Request for Reexamination of the Dickens patent 5,806,063 includes this Request and the associated Appendix entitled Appendix to Reexamination Request for Patent 5,806,063. The appendix includes claim charts applying the claims to various references as well as the following Exhibits:

REFERENCES

1 Shaughnessy U.S. Patent 5,630,118

2 Ohms, "Computer Processing of Dates Outside the Twentieth Century", *IBM Systems Journal*, v.25, #2, 1986, pp 244 et seq.

3 Japanese Published Application of Hazama, JP-5-27947, February 5, 1993

4 Japanese Published Application of Saka, JP-6-103133, April 15, 1994

5 Figure 2 of the application as filed

6 Amendments to originally filed claims 1 and 11

7 Exhibit A accompanying the application

8 Certificate of Correction.

This request identifies the prior art raising the new issue of patentability and then briefly describes the Dickens disclosure and the prosecution of the original application. Because the written description requirement of 35 USC § 112 plays an important part in the reissue claims, the metes and bounds of the requirement are briefly described. The four references relied on in this request are then described. Because of the number of overlapping claims requestor has provided a discussion of the way the claims are interrelated under the heading Construction of the Claims. Special attention is paid to Dickens' new argument, separately as applied to the patent claims and the added reissue claims. Thereafter a representative new issue of patentability is outlined with respect to each of the claims.

Requestor submits there is a new issue of patentability raised by any of the following:

a. Shaughnessy U.S. Patent 5,630,118 (Exhibit 1)

b. Ohms, "Computer Processing of Dates Outside the Twentieth Century", *IBM Systems Journal*, v.25, #2, 1986, pp 244 et seq. (Exhibit 2)

c. Japanese Published Application of Hazama, JP-5-27947, February 5, 1993 (Exhibit 3)

d. Japanese Published Application of Saka, JP-6-103133, April 15, 1994. (Exhibit 4)

Each of the foregoing references disclose subject matter which is related to at least one claim of the patent, the references were not considered during the prosecution, and each of the references was published more than a year prior to the filing date of the application. During the original prosecution, the only prior art rejection was withdrawn when the patentee purported to antedate the reference that had been relied on. Accordingly, <u>any</u> pertinent reference would raise a new issue of patentability with respect to the original prosecution.

MPEP §2242, indicates that a new question of patentability is raised so long as the same question was not considered during the original prosecution or during any concluded reexamination or reissue prosecution. Although the patent which is the subject of this request for reexamination is the subject of three reexaminations and a reissue application (all of which have been merged), none of those four proceedings has concluded. Accordingly, the issues raised in those four proceedings do not bar requester's reliance on the above identified references. MPEP §2242 provides that:

> "If the prior art patents and printed publications raise a substantial question of patentability over at least one claim of the patent, then a substantial new question of patentability is present, unless the same question of patentability has already been decided by (1) a final holding of invalidity by a Federal Court, or (2) by the office in a previous examination of the patent. A 'previous examination of the patent' is: (1) the original examination of the examination of the application which matured into the patent; (2) the examination in the reissue application that has resulted in the reissue of the patent; or (3) an earlier concluded reexamination."

The reexamination is not concluded until the issuance of a reexamination certificate, *In re Bass,* 65 USPQ 2d 1156, 1157 (CAFC 2002).

Dickens disclosure

The disclosure of the patent purports to describe (in five typed pages) a solution to the year 2000 (Y2K) problem. As described, the problem arises from the conjunction of the use of two digits to designate a year, and the transition from the 20^{th} to the 21^{st} century. For example, consider whether the year "01" is before or after the year "98". By using only two digits to identify the year, it is impossible to tell whether "01" refers to 2001 or 1901. The former is after 1998 and the later precedes 1998.

The specification describes the use of a six digit date format YYMMDD (where each Y represents a year digit, each M represents a month digit and each D represents a day digit). Assuming a database includes dates in the foregoing format, the patent describes a solution as follows:

> "A 10-decade window with a Y_AY_B value for the first year of the 10-decade window is selected, Y_AY_B being no later than the earliest Y_1Y_2 year designator in

> the database. A century designator C_1C_2 is determined for each date in the database, C_1C_2 having a first value of Y_1Y_2 is less than Y_AY_B and having a second value if Y_1Y_2 is equal to or greater than Y_AY_B. Each date in the database is formatted with the values C_1C_2, Y_1Y_2, M_1M_2, and D_1D_2." (column 1, line 63 to column 2, line 3)

In other words, given the database or collection of data which is restricted to a date range of no more than ten decades, any 2-digit year date representation can be precisely located relative to the century boundary by selecting a base year which is no larger that the earliest date represented in the database. Although not mentioned in the specification this selection requires knowledge of the actual years represented in the database – i.e., the selection cannot be made based solely on the last two digits of the year dates in the database. Having made the selection, to locate any year merely requires comparison of the 2-digit representation of the year in question (Y_1Y_2) to the 2-digit representation of the base year (Y_AY_B). If the representation is smaller than the base year (Y_AY_B), then the date must be beyond the century boundary, i.e., for Y2K, the year is in the 21st century. This follows because we chose the base year to be as early as the earliest year in the collection. Once we determine that the representation in question (Y_1Y_2) is smaller than the base year, then by definition the year which is represented must be beyond the century boundary. Conversely, if the year representation (Y_1Y_2) is greater than or equal to the base, then the year which is being represented is determined to be in the earlier century, i.e., the 20th century. The result again follows from the manner in which we chose the base year.

For example, assume the database has data corresponding to years from 1967 on. We select the base representation of 66 (alternatively, we could have selected 67, but no more than 67). Assume we retrieve the 2-digits "85" as the data to be tested. Since "85" is larger than "66" we determine that "85" represents the year 1985 and not the year 2085. As another example, assume that we retrieve the data "42". From the comparison between "42" and "66", we determine that "42" is less than "66". Consequently, we determine that the data "42" refers to the year 2042 and not to the year 1942*.

---

* It should be noted that any of the references cited in this request, or cited in any of the prior requests for reexamination as well as the references cited in the protest to the reissue, anticipates the use of windowing just described.

The summary of the invention indicates that existing "symbolic data representations are converted to a more useful form of symbolic date representations without the addition of new data fields, and in a manner that is performed automatically by the computer and requires no user input. The approach of the invention permits direct numerical sorting of dates." (Column 1, lines 51-56)

The specification further indicates that figure 1 illustrates a computer and various components. A database 26 is stored "in the memory 22 or on the mass-storage device 23". (Column 2, lines 46-47). This database has information in the form of the symbolic representations of dates. The specification continues that after applying the conversions just outlined above, there is a reformatting. The specification provides that "the symbolic representations of the dates in the database are reformatted with the values C_1C_2, Y_1Y_2, M_1M_2, and D_1D_2, step 36 of figure 2." (Column 3, lines 39-41). Reference to figure 2 of the patent indicates that the step 36 is not contained in figure 2 notwithstanding the statements in the specification. Reference to figure 2 as originally filed (see Exhibit 5) shows that step 36 in the process illustrated in figure 2 has been deleted.

The specification then describes the form of the reformatted date information, i.e., $C_1C_2Y_1Y_2M_1M_2D_1D_2$. In the paragraph at column 3, lines 48-55 the specification indicates that data in the format $C_1C_2Y_1Y_2M_1M_2D_1D_2$ may be sorted or otherwise manipulated and the manipulation may include "storing the dates and associated information back in the database" (column 3, lines 54-55). The specification concludes by referencing Exhibit A (column 3, lines 56-59) and referring to unidentified "modifications and enhancements" (column 3, line 61-column 4, line 4).

As will be explored in more detail below, there is absolutely no description of transferring data from a mass storage device to random access or electronic memory (volatile memory), executing the reformatting exclusively in random access memory and refraining from re-writing the database stored in the mass storage device (disk storage) as Dickens now argues is his invention. Contrary to Dickens' new argument, fig. 2 as filed (Exhibit 5) expressly describes "reformat dates in database", the specification (in the issued patent) calls for "storing the dates and associated information back in the database" (column 3, line 54), original claim 1 recited

"reformatting the ... date in the database" and claim 11 recited "reformatting each date in the database".

Prosecution

As filed the independent claim 1 required:

providing a **database** with **symbolic representations of dates stored therein** according to a format wherein M_1 M_2 is the numerical month designator, D_1 D_2 is the numerical day designator, and Y_1 Y_2 is the numerical year designator; .... and

reformatting **the symbolic representation of the date in the database** with the values C_1 C_2, Y_1 Y_2, M_1 M_2, D_1 D_2.

The only other independent claim in the application as filed, claim 11, was similar in requiring:

providing a database with symbolic representations of **dates stored therein** ....;

....; and

reformatting **each date in the database** ....".

On November 17, 1997, the Examiner issued an Office Action rejecting all pending claims on two different grounds. All claims were rejected as being anticipated by a prior art publication and all claims were rejected under 35 USC 112, 1st paragraph.

Subsequent to the rejection, the applicant purported to antedate the reference. The Examiner accepted this showing and removed the reference. Thus none of the claims even attempted to distinguish from that reference. The applicant also amended the claims and drawings to induce the Examiner to allow the application. In particular, figure 2 was amended (Exhibit 5), and claims 1 and 11 were amended (see Exhibit 6). Figure 2 was amended by deleting step 36 "reformat dates in database". Claim 1, which had recited "reformatting the symbolic representation of the date in the database" was amended to delete the phrase "in the database". Claim 11, which had recited "reformatting each date in the database" was amended to again delete the phrase "in the database".

More particularly, in the supplemental amendment of April 98 claims were amended to make it clear that some claims were NOT limited to subject matter in which the reformatted data was stored. At page 4 of that amendment Dickens stated:

> "Accordingly, independent claims 1 and 11 have now been amended so as to **not** require storage of the converted dates, thereby not imposing any requirements for new data fields." (emphasis in the original)

Since claims 1-15 have not changed since the April 1998 amendment, it appears, in view of Dickens' remarks in that amendment that all original claims, other than claims 9 and 14, read on methods which do not store converted data, i.e., methods which do not rewrite a database. Claims 9 and 14 cannot read on a method where a database is not rewritten since they positively call for the reformatted data to be stored into the database. But while most of the patent claims do not require rewriting of the database, nevertheless since the claims do not preclude rewriting the database it follows that not only claims 9 and 14 (which require rewriting the database by storing the reformatted dates) but ALL claims read on methods where a database is rewritten. Claims 9 and 14 read on such a method because they expressly require the converted dates to be stored. The other patent claims read on a database rewriting method simply because they are too broad to exclude that subject matter.

In the most recent response (Dec. 2002) Dickens now argues that his invention depends on **not** rewriting the database. In other words Dickens argues he is entitled to a patent because, whereas the prior art taught converting dates in the same manner as he does, the prior art failed to recognize the advantage of **not** rewriting the database. There are multiple fallacies in this argument. Three important fallacies are:

1. No such invention was disclosed until the Dec. 2002 response;
2. None of the patent claims are directed to such an invention; and
3. Many of the reissue claims are, likewise not directed to such an invention.

Because of the failure to disclose this invention, any claim which **is** directed to that invention runs afoul of the written description requirement of 35 USC 112, 1st paragraph.

WRITTEN DESCRIPTION REQUIREMENT

35 USC Section 112, first paragraph (112/1st) mandates three separate requirements. Particularly, the specification must include a written description of:

1. "the invention"; {the written description requirement}

2. "the manner and process of making and using [the invention] . . .so as to enable any person skilled in the art . . . to make and use [the invention]", {the enablement requirement}

3. "the best mode contemplated by the inventor of carrying out [the] invention." {the best mode requirement}.

To satisfy the written description requirement, "applicant must … convey to those skilled in the art that, as of the filing date sought, he or she was in possession of the invention. The invention is, for purposes of the 'written description' inquiry is whatever is now claimed." *Vas-Cath, Inc. v. Mahurkar*, 935 F.2d 1555, 1563-66 (CAFC 1991) (emphasis in original). "One shows that one is in 'in possession' of the invention by describing the invention, with all its claimed limitations. . ." *Lockwood v. American Airlines, Inc.,* 107 F.3d 1565, 1572 (CAFC 1997) (emphasis added).

The "written description" requirement is also clearly explained at MPEP 2163. The requirement can be summarized as follows (from MPEP 2163.01, at p. 2100-124):

> "If the examiner concludes that the claimed subject matter is not … described … in the application as filed, this would result in a rejection of the claims on the ground of a lack of written description under 35 U.S.C. 112, first paragraph …."

As indicated at MPEP 2163.05 the failure to meet the written description requirement commonly arises "when claims are changed", such as the new claims 16-76 introduced in the reissue application. Note *Ex parte Ohshiro*, 14 USPQ 2d 1750 (Bd. Pat. App. & Inter. 1989), cited at MPEP 2163.05II (p. 2100-127).

In other words, the original specification must provide a description of ALL of the features of ALL of the claims. Claim features which are not described in the specification evidence a failure to comply with the required "written description" of 35 USC 112, 1st paragraph. There are many examples of this failure in the newly presented claims.

Dickens argues (pp 15-16) that claims subject to reexamination should somehow be insulated from rejections under 35 USC § 112. However Dickens has overlooked the fact that he is prosecuting a reissue application. By virtue of 37CFR § 1.176 reissue claims are examined as ordinary application claims so rejections under 35 USC §112 are proper.

THE REFERENCES

Saka Japanese Published Application 06-103133, April 15, 1994 (Exhibit 4)

The Japanese published application is entitled "Method of managing date keys of a data file" [Title].* The reference is also addressed to Y2K [Object, 0005] and particularly describes a date key restoration unit which pre-pends to YYMMDD data, either "19" or "20" to represent a 20th century indication (19) or a 21st century indication (20) [0006]. The determination of whether the year is 20th or 21st century is made by comparing the year to a "threshold value". If the year is greater than the "threshold" then it is in the 20th century and if the year is smaller it is in the 21st century. [0009] Thereafter, the data is sorted [0012]. The table which is Key file 3 and Fig. 2 make it clear that the method of the reference includes revising the two-digit years in the database so that 4 digit year data is provided.

The reference's "threshold" parameter corresponds to Dickens' Y_AY_B. This parameter also establishes the extent of the "window". The reference acknowledges the 10 decade limitation on the logic by noting:

* The bracketed information is keyed to the various parts of the specification such as [Title], or numbered paragraphs such as [005].

> "In general, if the minimum value of the year data in the 20th century is 'n_0n_1', and the maximum value of the year data in the 21st century is 'n_2 n_3', for the threshold value, a suitable value between 'n_0n_1' and 'n_2 n_3' is employed." [0011]

The "threshold" (Y_AY_B) is between the minimum value of the year data in the 20th century (n_0n_1) and the maximum value of the year data in the 21st century (n_2 n_3) and so there is no 20th century value lower than the threshold, just as expressed in the Dickens claims.

Dickens argues that the references do not show writing date information for an entire database into electronic memory, using windowing to determine the proper century for each date in the database, reformatting the date information using the century information and then sorting the date information properly, based on the reformatted data. However this patent has just that disclosure. Fig. 2 shows a date database 1 in which year data is restricted to two digits. Fig. 2 also shows a date key restoration unit 4. Unit 4 compares the 2- digit year value to the "threshold value" and appends the century indicator (19 or 20) to make a four digit year value [0009, 0014-0015]. The key file compilation unit 5 writes into a "working region of memory ... and compiles a key file 3" with all the date keys arranged in a particular order, "for example, in ascending order by performing sorting processing" [0016]. Because of Y2k the data in file 1 may not be properly ordered, see [0003]. The solution is described above, i.e., pre-pending a century indicator based on whether or not the two digit year data is greater or not greater than a "year threshold", see [0006] and [0007]. Moreover the processing, as described in [0015] is "executed for the successive records of data file 1 and terminates when **the last record is processed**". (emphasis added) In other words the reformatting and sorting occurs for each record in the file. Note that the compilation unit 5 writes data, after reformatting, into a working region of memory. Thus data from the database 1 is reformatted and written to electronic memory just as Dickens **now** argues his invention was intended to operate. Whereas the Saka patent publication disclosed this operation in April 1994, Dickens did not reveal this operation until his response to the rejection in Dec. 2002.

Ohms (Exhibit 2)

The 1986 Ohms publication is also directed at date processing occasioned by the end of the 20th century, e.g., the Y2K problem. Ohms describes conversions between different date

formats. While he devotes several pages to explaining the Lilian format, at p. 248, under the heading "Accommodating systems support" he indicates that:

> "The conversion of isolated files to new date formats presents a rather trivial problem."

And on the next page he indicates that:

> "[A]s systems are maintained or replaced, it would be practical to implement full information date formats."

The fact that he contemplates applying these conversions to databases using YYMMDD data to express dates is readily identified by the table on page 247. The table indicates that Ohms describes a function to convert a "short Gregorian" date (having the format YYMMDD). Ohms succinctly describes the subject matter claimed by Dickens in the first full paragraph of the right hand column on page 248. Ohms states:

> "However, it may be necessary to provide a conversion function that receives a definition of the implied century as a parameter. An excellent way to do this unambiguously is to specify a year as the desired starting point of a 100-year range. For example, if the starting year for the range is specified as 1925, dates with year digits of 25 through 99 would be between 1925 and 1999, and dates with year digits of 00 through 24 would lie between 2000 and 2024."

Ohms emphasizes the caution against using this procedure where the date range spans more that 100 years by indicating (at page 249 in the left hand column) "Where systems contain dates that span a range of more than 100 years, the century must have already have been carried. In the rare event that this is not true, immediate conversion is unavoidable". Ohms like Dickens works with a 10-decade window (10-decades is identically 100 years). It should be emphasized that the dates within Ohms' 100 year range typically fall into two different centuries (just as in Dickens). The Ohms "starting point" (which is the same as the Y_AY_B parameter of Dickens) *defines* a 10-decade window (or 100 year range). It should be noted that even in lines 28-38 on p. 248 Ohms is still referring to the100 year range (which is identically the Dickens 10-decade window). Ohms, in the same passage, refers to a "beginning date" (which is the same as the "starting point") "set eighty years prior to the current systems date". This sentence relates the "beginning date" or "starting point" to a current date (the current systems date). This duration (eighty years in the text) has *nothing* to do with the duration or extent of 10-decade window (or 100 year range). In this passage Ohms relates the 100 year window to the present, so as to

indicate how far into the future the window extends. The Ohms example in which the start year is eighty years in the past also means that the database accommodates data for 20 years into the future (since the total span is 100 years). Any reference which comments on this relationship is more comprehensive than Dickens because *Dickens never mentions this relationship.*

Dickens attempts to limit the reach of Ohms to the single idea of converting a database to a Lilian format. Dickens suggests that the conversion function (p. 248 of Ohms) is described only for the purpose of data input to a file already converted to Lilian format. There is no reason one skilled in the art would read Ohms in such a limited way. In the paragraph immediately after describing the "conversion function" (at p. 248) Ohms mentions the "conversion of isolated files to a new date format", and notes that this "presents a rather trivial problem." This very subject matter represents the anticipation of Dickens. Just because Ohms characterizes the conversion as "trivial" is no reason to ignore the teaching.

Shaughnessy (Exhibit 1)

The Shaughnessy patent relates to computer systems that perform date operations on date fields spanning a century boundary. While the title is generic, the text indicates that it is Y2K which is the genesis of the patent. Shaughnessy describes the modifications to computer systems so that date operations can be performed correctly even when processing dates after December 31, 1999, e.g., Y2K. The data formats that are employed in accordance with Shaughnessy are found in a table attached as an appendix, see column 18. As indicated in column 18, there are several different formats that are represented as "YYMMDD". The appendix notes that for formats B, F and S (all of which are YYMMDD) that "the date cycle is 100 years". The general sequence of operations described by Shaughnessy is shown in figure 2 where the requested date operation is performed only after certain precursor steps are performed. The precursor steps include determining the current date, determining the end of the 100-year cycle and determining two possible century values. As shown in figure 4, the end of the 100-year cycle can be determined either based on the current date or based on the system installation date. In either event, once the end of the 100-year cycle is determined, the system derives two dates separated by a period of 100 years (10 decades).

Figure 5 shows how the two possible century values are determined. In particular, the later of the two centuries is determined as the century of the date at the end of the 100-year cycle, and the earlier century is the century proceeding the later century.

Figure 7 shows how the century value (this corresponds to C_1C_2 of Dickens) is assigned. Assignment includes a comparison between the date representation (e.g., YY) with a "end of the 100 year cycle". The text indicates that:

> "If the date is less than or equal to the end of the 100 year cycle date, the CENTURY2 value is assigned to the date (box 64). If the date is greater than the end of cycle date, the CENTURY1 value is assigned to the date (box 66)". Column 7, lines 9-13.

Note this Shaughnessy century determination is identical to the Dickens century determination. This identity is apparent by equating Dickens' Y_AY_B with the Shaughnessy "end of the 100 year cycle date".

After the century designator (this is CENTURY1 or CENTURY2) is assigned the date is reformatted to the format YYYYMMDD (see column 6 line 65). Of course, the first two of the Y digits represent the assigned century. This is identical to the format ($C_1C_2Y_1Y_2M_1M_2D_1D_2$) of Dickens claim 11.

While the Shaughnessy specification comprehends several embodiments, the embodiment which deals with a C1 date format is limited to date spans of 100 years or less. This is apparent from the determination of the "end of the 100 year cycle" and the fact the date ambiguity is resolved by determining which of two possible century values is appropriate.

Shaughnessy indicates (column 1, line 26 and column 8, line 34 – column 12, line 20) that computers typically compare dates. Indeed, sequentially comparing dates is also referred to as sorting and therefore in these passages Shaughnessy teaches that dates processed in accordance with the procedure just described can then be used for sorting purposes. Alternatively, the patent owner has admitted that sorting after Y2K correction is part of the prior art (Response, p. 14).

While the impetus for the Shaughnessy patent was Y2K (the problem occasioned by the transition from the 20th to the 21st century) it should be clear that the Shaughnessy specification

also describes solutions applicable to analogous but not necessarily identical problems. The Shaughnessy specification describes (1) maintaining a database unchanged (1/60-2/5) even though the data could be misinterpreted and (2) the alteration of the program logic by the addition of a subroutine. The subroutine allows the selection of the appropriate date. Reference to the Appendix (col. 18-19) indicates that Shaughnessy contemplated one data format in which the date "cycle" was 100 years (the C1 format) and another data format in which the date "cycle" was 10,000 years (C2). Shaughnessy describes that using a two digit year data format, 100 year cycle data is misinterpreted at the turn of the century (that is the reason it is necessary to determine whether a date is CENTURY1 or CENTURY2). Using four digit year data there is a similar problem in 10,000 year cycle data, i.e., when there is a transition from the year 9999 to the year 10,000. While the Shaughnessy specification may apply to databases with C1 type data, as well as database with C2 type data, there is nothing in the patent to suggest mixing 100 year cycle data with 10,000 year cycle data in one database. It should be clear from the description of "windowing" contained in Ohms (1986), that by the 1994 filing of the Shaughnessy application the art was well aware of the 100 year limit associated with two digit year data. Consequently the Shaughnessy description is quite adequate on the question of limiting data ranges to 100 years when interpreting data with two digit year data. The first Shaughnessy flowchart has a function to determine the "end of 100 year cycle" – this discloses the 100 year limit of year data to those skilled in the art as of 1994.

Shaughnessy, like some of the other references also relates the 100 year window to the present. This relationship is determined by the "number of years of future dating" (6/10). Shaughnessy also provides another degree of freedom to the user in that the "end of the 100 year cycle" may be updated (6/13). In both respects Shaughnessy goes beyond the Dickens specification in that *Dickens never mentions either parameter.* Neither of these parameters ("number of years of future dating" or the ability to update the "end of the 100 year cycle") bears on the manner in which Shaughnessy anticipates the Dickens 100 year window or the manner in which the correct century designator is determined. To be sure, Shaughnessy describes using the "end of the 100 year cycle" while Dickens uses the beginning of the 100 year cycle. However, for any given window these two numbers differ by unity. Consider a Dickens window defined by Y_AY_B of 50. This window extends from 1950 to 2049. The corresponding Shaughnessy "end of the 100 year cycle" is 49, and it defines the same window, from 1950 to 2049.

Hazama Japanese Patent Publication of Hazama 05/027,947 (Exhibit 3)

This Japanese patent publication is also directed at Y2K. The translation (page 2) notes, the purpose is:

> "To guarantee the year order, even for years after 2000AD, with the current file format, even when the year is managed by the last 2 digits of the date in digital files."

The Y2K problem is described at page 3 of the translation where the text indicates that systems using the last two digits of the years to indicate dates do not take into consideration years after 2000. The text continues that:

> "When ascending/descending order is handled by processing that evaluates magnitude and by sort/merge processing using normally numbered years, their relative magnitudes are represented by formula 1
>
> $$19\underline{99} > 19\underline{98} > 20\underline{01} > 20\underline{00}.$$"

In other words prior art systems focusing only on years expressed as two digits will, when sorting the four years 1998-2001, produce a result which indicates that the latest year is 1999, which is preceded by 1998 which is preceded by 2001 which is preceded in turn by 2000. This clearly is incorrect. On page 4, the text indicates that the invention provides a method of guaranteeing the proper year order and indicates that in general when there are data present that indicate years in the 20th and the 21st century, the data code that represents dates in the 21st century is replaced by another code so the year order will be maintained. The manner in which this is accomplished is described at the bottom of page 4. A module 10 is activated to effect preprocessing in order to handle the year calculations. The text indicates:

> "Note that, in module (10), a range of the last 2 digits for which code transformation will be performed are specified in advance. Replacement involves numbers for years in the 2000's, where the last two digits are smaller than the smallest number in the last 2 digits in the years in the 1900's. For example, when data in file (6) for years AD begin with the year 1973, the last two digits are replaced using 00 (year 2000) for 72 (year 2072). The present application example is an example where there are data from year 1960 in file (6), such a range is specified so that the last 2 digits will be transformed to codes 00-59".

In other words, year data which is "smaller than the smallest number in the last two digits in the years in the 1900s" are recoded to indicate dates in the 21st century. Of course Dickens called recoding, reformatting. What the translation refers to as "the smallest number in the last two digits in the years in the 1900's" Dickens labeled Y_AY_B. The two examples employed

illustrate this processing is necessarily limited to 10-decade ranges. The text indicates when the input file has year dates beginning with the year 1973 (Y_AY_B=73), the data is recoded to indicate dates up to and including 72 lie in the 21st century. Similarly, when the lowest dates in the input file is 1960 (Y_AY_B=60) dates 00-59 are recoded (or reformatted) indicate they lie in the 21st century.

Dickens again asserts that this reference does not disclose reformatting more than one or two dates in the file. Dickens has not read the text closely enough. After describing how a single record is modified to reflect the proper century, the text continues:

> "In addition to this, the next records are input." [see 0015],

the text concludes:

> "By replacing code so that the relative magnitudes [note the use of the plural] of the years [note use of plural] are correctly evaluated, the effect is that year order will be guaranteed without changing data file record length or block length, and further, without modifying programs." [see 0025]

As described the invention is directed at "a method of guaranteeing year order ... with ... "sort/merge processing using programs for digital files ...". [see 0001] Of course no sort will be correct unless all dates are reformatted.

Construction of the Claims

The 76 claims which are now pending can be divided into two distinct groups. The first group of claims is directed to a method reciting the steps of providing, selecting, determining and reformatting with variations in one or more of these clauses. The claims in the first group are claims 1-33, 66-69 and 72-76.

Different from these claims is another set of claims that call for converting by windowing and running a program. These claims include claims 34-65, 70 and 71.

Dealing with the provide/select/determine/reformat category of claims, the independent claims have variations, mostly in their re-format clause but some in the providing clause.

More particularly, the reformat clause in claim 1 specifies that the reformatted format has CC, YY, MM and DD data.

On the other hand, the reformatting clause in claim 11 is somewhat narrower. This reformat clause specifies the format which is produced is CCYYMMDD. In addition, claim 11 recites a sorting step.

The reformatting clause of claim 16 is similar to the reformatting clause of claim 1 except that it specifies that the reformatting does not require a new data field.

The reformatting clause of claim 26 is similar to that of claim 16 and goes on to also specify that the reformatting is intended to facilitate collectively further processing.

The reformatting clause of claim 31 is slightly broader than the reformatting clause of claim 16.

Claim 32 is similar to claim 31 but adds a sorting step.

The reformatting clause of claim 33 is similar to claim 1 but it specifies that the reformatting occurs "without changing".

Claim 66 is similar to claim 26 with several exceptions. Claim 66 recites the reformatting of "each symbolic representation" operates on "a date in a portion of the at least one date field" and that the reformatting is repeated until each date is reformatted. Whereas claim 26 refers in the "providing" clause to "the symbolic representations", claim 66, on the other hand refers, for the first time to "the symbolic representation" in the claim in the "reformatting" clause. In other words there is no predicate for the reference back.

Claim 67 is similar to claim 66 with the exception that the reformat clause of claim 66 refers to YY, MM and DD date data, while claim 67 refers only to YY date data.

The reformatting clause of claim 68 is similar to that of claim 31 but indicates that the further processing includes running a program.

Claim 69 is similar to claim 68 except that it adds a step of sorting.

Claim 72 is similar to claim 1 except that it adds a recitation concerned with collectively further processing.

Claim 73 and 74 are respectively broader versions of claims 1 and 11.

Claim 75 is similar to claim 16 except that it uses the phrase "pivot date",

Finally, claim 76 is similar to claim 26 except that it uses the phrase "pivot date".

Claims dependent on this group of independent claims include claims 2-10, 12-15, 17-25 and 27-30. Different ones of these dependent claims are directed at one or more of the following concepts:

a. the window includes the year 2000;
b. the century designation includes 19 or 20 representing the 20^{th} or 21^{st} century;
c. adds a step of sorting;
d. specifies that the format is CCYYMMDD;
e. indicates that the sorting is a numerical order sort;
f. converting from some different format to the YYMMDD format;
g. Y_B=0;
h. there is a storing operation after the reformatting;
i. there is manipulation after reformatting; and
j. there is manipulation after sorting.

The foregoing claim recitations raise a number of issues of claim interpretation.

Claims 16-32, 34-59, 68-71, 75 and 76 use the phrase "without the addition of any new data field".

The specification describes how the reformat is accomplished by adding the digits CC to the existing year digits YY and this produces the four-digit year CCYY. Thus, prior to the reformatting, the database presumably includes, at most, date data fields for years (YY), months (MM) and days (DD). After the reformatting there are still, at most, three data fields, the year

data field has been expanded (to include CCYY) and the month and day data fields have not changed. Thus, this is indeed reformatting "without the addition of any new data field". Note, however, that it does require the expansion of the year field from YY to CCYY. As will be apparent, many of the references cited against these claims likewise expand the year field from two digits (YY) to four digits (CCYY). Saka and Shaughnessy are two examples. It goes without saying that inasmuch as the claims and the specification of the patent describe the same year data field expansion as found in the references, the claim phrase "without the addition of any new data field" does **not** distinguish over these references.

Claims 16-33 and 72 include a phrase indicating that the reformatting is "in order to facilitate collectively further processing". The specification of the Dickens patent describes that the reformatted data can be numerically sorted whereas it could not be numerically sorted prior to the reformatting. To this extent the reformatting facilitates processing of the data as a whole, e.g., "collectively further processing". As is apparent, the references also describe a reformatting which adds century digits to the year digits. It is important to understand that these claims do not specify the further processing which is facilitated, merely that the reformatting facilitates further processing. Thus, to the extent that a reference modifies a two-digit year date to include a four-digit year date, that operation too facilitates collective further processing. Thus this phrase of the claims does not distinguish from the references such as Saka, Hazama or Shaughnessy.

Claim 33 specifies that the reformatting occurs "without changing any of the symbolic representations of a date in the database during the reformatting step". The Dickens patent describes that the reformatting is implemented by changing YY to CCYY (3:38-43). This indeed is a change in the symbolic representation of a date in the database which occurs during the reformatting step. There is no other disclosure in the Dickens specification. In particular, there is no written description of how reformatting is accomplished "without changing any of the symbolic representations of a date in the database during the reformatting step". It follows, therefore, that there is no written description in the Dickens patent specification of an invention including the quoted subject matter.

In the latest reply, Dickens now argues the follow scenario:

1. The database, including the two-digit year date data, is contained in a legacy database;

2. this information is transferred somewhere else (not specified);

3. the reformatting occurs in the unspecified location; and

4. the legacy database is **not** rewritten.

Then if we limit the term database to refer only to the legacy database (why?) and refrain from using the term database to refer to the information written elsewhere, such as in electronic or random access memory, then:

we can assert that reformatting occurs without changing a date "in the database" (where the database is limited to refer only to the legacy database presumably found in a mass storage device or permanent memory).

Dickens does not argue that this tortured sequence is described in the specification. Rather, he argues that anyone of skill in the art would imply this sequence from the words which are actually found in the specification (pp 20-21). In other words, although Dickens acknowledges that the words and drawings necessary to describe this sequence are not found in the application, nevertheless, Dickens argues, from the words and the drawings which **are** present one skilled in the art would derive this information, i.e., Dickens argues the description is **inherent**.

Two difficulties with this argument are (1) that the specification **contradicts** it, and (2) that the original drawings **contradict** it. For one thing, the specification does not distinguish between the database in a mass storage device or in random access memory, see column 2, lines 44-48. In other words the data in electronic memory is also the database (2:44-48). Secondly, the specification and claims expressly recite that once the dates have been reformatted the operation includes "storing the dates and associated information back in the database". Indeed,

this operation is recited in claims 9 and 14 of the patent as well as in claims asserted in the very same reissue application.

Dickens argues that during the prosecution this particular scenario was described in detail, see the response pp. 24-25. However, the written description required by the statute must be in the application as filed - it cannot be added later either by amendment or by argument.

In conclusion, Dickens admits that this tortured sequence is not expressed in the specification, and since the only the information expressed in the specification is directly contrary, it cannot be said that this operation is inherent. Lacking either express or inherent description, it is clear that the claim lacks the required written description for the "without changing" subject matter.

Claims 9 and 24 recite the step of "storing the symbolic representation of dates and their associated information back into the database".

Claims 14 and 29 recite "storing the sorted dates and their associated information back into the database".

The Dickens specification does not distinguish between the database stored in the mass storage device or in a random access memory (see column 2, lines 44-48). To the extent that references, such as Saka Japanese Patent publication or the Hazama Patent publication describe reformatting date data in a table stored in random access memory, the requestor submits that these teachings meet the requirement of storing dates back in the database. Alternatively, references, such as Ohms or Shaughnessy, which teach rewriting a database, that is recording in a mass storage device, the reformatted dates, also meet these recitations.

Claims 20-21 call for "reformatting … separately from the database". As was the case for the "without changing" clause from claim 33, the subject of "reformatting … separately from the database" is simply not described in the Dickens specification. As has been noted, the Dickens specification does not distinguish between a database in a mass storage and a database in random access memory (2:44-48). Since there is no description in the Dickens patent of reformatting separate from the database, these claims lack the required written description. It is possible to perform some metal gymnastics to make this clause consistent with the specification.

This is accomplished by implying that the reformatting operation occurs in an (undescribed but implied) arithmetic logic unit (ALU) which, at the time, stores only a single one of the dates and thus could be said to be separate from the database whether or not the database is stored in a mass storage device or in random access memory. This "reading" of the clause, however, does not distinguish the clause from any of the cited prior art – which presumably include the same undescribed (but implied) ALU.

Claims 16-30, 32, 34-67, 69-71, 75 and 76 refer to a "pivot year". This phrase does not appear in the specification or claims of the patent. If we assume that the phrase is a pseudonym for Y_AY_B, then the claims can be reconciled with the specification. If an attempt is made to give this phrase any other meaning then, the claims are subject to invalidity based on lack of a written description.

Claims 32 and 69 call for "sorting the dates in the form CCYY". There is no description in the Dickens specification of sorting on CCYY. The specification discloses sorting dates in the form CCYYMMDD. It goes without saying that these two forms are different. Dickens argues that the form CCYY is generic to CCYYMMDD. Requestor disagrees. This is not a case where the specification describes an invention which may incorporate any one element from a genus. Such disclosure will support claiming a genus. In this case, however, CCYY and CCYYMMDD are two different species; neither is generic to the other. The specification discloses sorting on CCYMMDD, that specification does not describe or enable support for sorting on CCYY. Claims 32 and 69 fail to comply with 35 USC 112 since the specification does not provide a written description for "sorting the dates in the form CCYY".

The other group of claims (34-65, 70 and 71) have either two or three step claims depending on how they are read. The first step of claim 34 includes at least "converting each of the symbolic representations of dates stored in the at least one date field of a database to a symbolic representation of each of the respective dates that does not create the ambiguity". The step of converting then specifies that the converting is accomplished "by windowing the symbolic representation of each of the respective dates … against a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database, without the addition of any new data field to a database for the purposes of such windowing and converting".

Claim 34 concludes by calling for "running a program collectively on each of the converted symbolic representations of each of the respective dates to sort or otherwise manipulate the dates represented by the converted symbolic representations, separately from the date data symbolic representations contained in the at least one date field of the database."

The other independent claims in this group, 60-65, 70 and 71 include the following variations on the subject matter of claim 34. Claim 60 omits the recitation that the windowing occurs "without adding a new data field" but specifies that the windowing occurs "without modifying any of the symbolic representation of dates in the at least one date field". In addition, the running of a program clause omits the characterization that the execution is "collective".

Claim 61 is similar to claim 60 except that it adds "collectively" to the running of a program clause.

Claim 62 is like claim 34 and further specifies "storing the converted symbolic representations separate" from the "database" and omits the recitation that running the program occurs "collectively".

Claim 63 is like claim 34 but again adds the recitation of "storing the converted symbolic representations separate" from the "database".

Claim 64 is like claim 60 but specifies "storing the converted symbolic representations separate" from the "database".

Claim 65 is like claim 64 but adds the recitation that running a program is effected "collectively".

Claim 70 is similar to claim 34 but rather than specifying that the pivot year is from the database, it specifies that the pivot year is less than or equal to the earliest date in the database. Like claim 34, claim 70 specifies that the windowing occurs "without adding a new data field" but it also specifies that the windowing occurs "without modifying".

Finally, claim 71 is like claim 70 except that it specifies storing converted representation separate from the database.

The only dependent claims in this group are claims dependent on claim 34. This includes claims 35-59. These claims add one or more of the following concepts to their respective parent claims:

a. opening a database;
b. collectively sorting;
c. collectively manipulating;
d. collectively sorting "according to a different data field";
e. collectively manipulating "according to a different data field";
f. program manipulates data; and
g. convert at least substantial portions.

Claims 34-65 specify that the windowing occurs relative to "a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database". This clause requires that the pivot year is to be a year which is also an entry in the database. As noted above, the phrase "pivot year" does not appear in the specification but if we assume it is a pseudonym for Y_AY_B, the claim does not run afoul of the written description requirement on this basis. Note how specific the language is, "a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database" - in other words it is not enough that this "pivot year" is within the range of the data in the database, the language is specific that the pivot year is represented "by one of the ... dates as stored in the .... database". There is no suggestion of this subject matter either in the specification or claims of the patent. Thus, for this reason claims 34-65 are not supported by a written description as required by 35 USC 112, 1st paragraph.

Claims 34-65, 70 and 71 contain three additional clauses of interest. These can be short handed as "sorting separate" from the database (claims 34, 60-65, 70 and 71), "storing separate" from the database (claims 62-65 and 71) converting "without modifying" (claims 60-61, 64-65 and 70).

Claims 34-61 call for "running a program … on each of the converted symbolic representations of each of the respective dates to sort or otherwise manipulate the dates …

separately from the date data symbolic representations contained in the at least one date field of the database." There is no description of sorting "separately from date data symbolic representations contained in the at least one date field of the database". If this clause means that an undescribed but implied arithmetic logic unit (ALU) which implements the sorting is separate from a random access memory or mass storage device in which the information making up the database is stored, then the clause will not run afoul of 35 USC 112. At the same time, however, the clause will not distinguish this claim from any of the cited art which describes sorting after adding century information to year date information in a database. If we can "construe" the Dickens specification to include an undescribed ALU to reconcile these claims with the otherwise lacking description, then we should be able to imply the very same (undescribed) ALU in any of the references which adds the century indicator and then performs a sort. These descriptions, for example, are found at least in the Saka Japanese publication or the Hazama patent publication.

Claims 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57 and 59 depend directly or indirectly from claim 34 and call for the step of "opening the database", where the "database" is "a database utilizing symbolic representations of the dates stored in the at least one date field of the database" as recited in parent claim 34. There is no description in the Dickens specification or drawings of "opening the database" as recited in each of claims 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57 and 59. Dickens does not point to any part of the specification which describes "opening the database" in the response of Dec. 2002. Rather Dickens notes that Exhibit A (which was attached to the application as filed) references opening the tools database (p. 63). The Exhibit A which Dickens refers to is attached hereto as Exhibit 7. This document does not support Dickens' reliance. What it actually says is:

```
10  open structure tools:name 'otms_src_dir:tools'
    open #2  :  name  'last_inv.dat'  ,  access  output
```

There is no indication that either of these murky statements refer to opening a database much less the database having "symbolic representations of the dates" as called for in the claim. If one of these statements was intended to teach that opening a database was a step of the Dickens invention, one or two words to that effect would have sufficed to make that clear.

Rather we have Dickens now, 7 years later attempting to read into this obscure passage the description necessary to support these claims - this is too little - too late. There is no written description for the step of "opening" a database which is found in claims.

Alternatively Dickens says opening a database is "inherent". (Response p. 63 and paragraph 178 of the Declaration). Requestor asserts that if it is inherent in the Dickens disclosure it is likewise inherent in any of the cited references.

Claims 40, 41, 52 and 53 depend directly or indirectly on claim 34 and require "sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field". Claim 34 refers to "representing and utilizing dates stored in at least one date field of a database which are in a format that creates ambiguity between dates in each of a pair of adjacent centuries". Thus the "one date field" of the database stores dates which may create an ambiguity. Since Dickens never specifies how many date fields are used (one date field to store CCYYMMDD or three date fields, each storing one of CCYY, MM and DD) we cannot be certain about the precise content of the "one date field". Regardless, it is clear that the one date field is either CCYY or CCYYMMDD. The dependent claims 40 and 41 then specify sorting "according to a different data field contained in the database from the at least one date field", i.e., sorting on something other than CCYY or CCYYMMDD. The entire Dickens disclosure of sorting is contained in line 10 of the Abstract, column 2, lines 18-21 and column 3, lines 46-55. These passages describe sorting on CCYYMMDD. In short there is absolutely no description of "sorting the converted symbolic representations according to a different data field contained in the database from the at least one date field" as recited in these claims. These four claims are unpatentable for the lack of a written description of the claimed subject matter.

Claims 42 and 43 have even less support. These claims depend either directly or indirectly from claim 34 and recite "manipulating the converted symbolic representations according to a different data field contained in the database from the at least one date field". In other words these claims call for some manipulation which depends on a data field other than either CCYY or CCYYMMDD. The entire disclosure of manipulating data in the Dickens patent is found in line 2 of the Abstract, column 1, lines 7 and 39 and column 3, lines 51-53. Other than mentioning that the data may be manipulated there is no indication of "manipulating

the converted symbolic representations according to" any data field much less "manipulating the converted symbolic representations according to a different data field contained in the database from the at least one date field" which is found in these claims. Both claims are unpatentable for the lack of a written description of the claimed subject matter.

Claims 62-65 and 70-71 contain a recitation similar to that of claims 34-61, that of "running a program … on the stored symbolic representations to sort or otherwise manipulate data in the database … separately from the symbolic representations of dates contained in the at least one date field of the database." As noted above, there is no description in the Dickens patents of sorting "separately from the symbolic representations of dates contained in the at least one date field of the database." This clause can only be reconciled with the Dickens patent by considering that it highlights the fact that sorting is implemented in an (undescribed but implied) ALU which is separate and apart from the random access memory or mass storage device which contains the database which is the subject of the sorting operation. Only with this construction would this clause comply with the description requirement of 35 USC 112. However, with this construction, the clause does not distinguish from any of the Y2K references cited herein which include a sorting step, such as the Saka Japanese patent publication or the Hazama patent publication for reasons similar to those mentioned in connection with claims 34-61.

Claims 62-65 and 71 call for "storing the converted symbolic representation separate from the at least one date field in the database". The Dickens specification indicates that the database is stored either in random access memory or in a mass storage device (2:46-48). The Dickens specification also indicates that once reformatted or converted, the reformatted data may be stored back into the database (column 3, lines 53-55, and original claims 9 and 14). There is no indication in this description that the converted information is stored "separate from the at least one data field of the database" as is recited in claims 62-65 and 71. Accordingly, these claims lack a written description as required by 35 USC 112, 1st paragraph.

Claims 60-61, 64-65 and 70 call for "converting each of the symbolic representations of dates … by windowing … without modifying any of the symbolic representations of dates in the at least one date field of the database for purposes of such windowing and converting". The specification of the Dickens patents describes that the converting or reformatting is implemented by adding the CC designation to the preexisting YY year date information. This operation

clearly modifies the representation. The Dickens patent indicates that the database, from which the YY year information is obtained, may be stored either in a mass storage device or in a random access memory (2:46-48). Contrary to the subject matter of this clause then, the specification describes that the converting occurs by changing the symbolic representation YY to the symbolic representation CCYY. It goes without saying that the change from YY to CCYY occurs in data stored in random access memory; it may even occur in the information stored in the mass storage device. It is beyond question, however, that there is no description of performing a conversion or reformatting "without modifying" the symbolic representation. The whole act of the formatting operation is to make a change or to modify the year date YY. In view of the fact that there is no description of effecting a conversion "without modifying" these claims are defective in that they are not supported by a written description as required by 35 USC 112, 1st paragraph. If these claims can be reconciled with the specification (such as by restricting the application of the term "database" to a mass storage device - contrary to the Dickens specification - and then ignoring the prescription to store the reformatted dates "back in the data base" {3:54-55}) then, with this "construction" these claims fail to distinguish from cited references, such as the Saka or Hazama patent publications or the Shaughnessy patent. Saka is actually a better description of these claims than the Dickens specification. Dickens does not distinguish between a database in a mass storage device or in electronic memory (2:46-48). Consequently when he describes reformatting the "symbolic representations" there is no suggestion that electronic memory alone is reformatted while the mass storage device is not. Saka, on the other hand, describes both a database 1 having only two digit year dates and a separate key file which is compiled in a "working region of memory" [0016]. The key file 3 has the converted or reformatted year dates, and so it could be properly said that Saka describes converting "without modifying" the database 1, if that is what is claimed.

Claims 16-33, 66, 67, 75 and 76 indicate that the reformatting is "in order to facilitate collectively further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates". The "in order to facilitate" language does not impose a further limitation on the claim. In other words, these claims are made up of several steps; claim 16 is an example. It is made up of four steps, the first step is "providing …", the second step is "selecting …", the third step is "determining …" and the fourth step is

"reformatting … in order to facilitate …". The aforesaid claims (16-33, 66, 67, 75 and 76) do not require any "further processing". These claims merely indicate that once reformatted, further processing is possible. Accordingly, references (such as those cited herein) which implement reformatting also "facilitate further processing of the reformatted symbolic representations" as claimed.

Claim 68 is a further variation on the foregoing. Claim 68 calls for "reformatting … in order to facilitate further processing with a reformatted symbolic representations of each of the symbolic representations of each of the dates, by running a program on the reformatted symbolic representations of each of the dates." The Dickens specification describes the reformatting, and it may be implied that the reformatting does facilitate further processing. The Dickens specification also describes that sorting can be accomplished subsequent to reformatting. It is also known (although not stated in so many words in the Dickens specification) that sorting is accomplished by running a program. Thus, the "running a program" clause of claim 68 may be considered to be supported by a sorting operation. In like fashion, references which teach the windowing process followed by a sorting, on dates which have been augmented by century indications, such as the Saka Japanese patent publication or the Hazama patent publication, likewise teach "reformatting … in order to facilitate further processing … by running a program …" as claimed in claim 68.

Claim 72 has a somewhat related recitation. Claim 72 calls for "reformatting … prior to collectively further processing information contained within a database associated with the respective dates." It is important to understand that the last clause of the claim calls for "reformatting …". The "further processing" only occurs subsequent to the "reformatting" and hence, is not an element of the claim. In any event, the "further processing" is supported in the Dickens specification by the disclosure of sorting. Hence any reference which discloses reformatting, e.g., adding the CC century designator to the year digits YY and then proceeds to perform a sorting operation, meets this clause of the claim. Such references include, for example, the Saka Japanese patent publication and the Hazama patent publication.

Claims 66 and 67 are subject to a further defect. Both claims have in common a "reformatting" step which recites, in part "reformatting the symbolic representation of **each** symbolic representation of a date in a portion off the at least one date field in the database"

(emphasis added). Thereafter the claims have a further step of "**repeating** the step of reformatting until each symbolic representation of a date in the at least one date field has been reformatted" (emphasis added). If the "reformatting" step reformats **each** date representation (as it must since that is expressly recited) query: what does the "repeating" step reformat? Since the Dickens disclosure describes only reformatting the dates (see column 3, lines 48-50) the repetitive reformatting claimed in claims 66 and 67 is unsupported by a written description and hence these claims are unpatentable for this reason. Alternatively the claims could be considered defective under 35 USC § 112, 2nd paragraph, since the "reformatting" step is effective to reformat "each" date there is no function to the repetitive reformatting and thus the claim scope is indefinite.

DICKENS' NEW ARGUMENT

Surprisingly, after the original prosecution (1996 through 1998), responding to three separate reexamination requests, filing a reissue application (2000), responding to two office actions (2001 and 2002), Dickens, for the first time, raises a new argument. He acknowledges that the prior art has employed windowing in two different ways. One way was the one-time modification of the entire database to reformat the two-digit year field with four-digit year fields. (December 2002 reply, page 18)

In addition, Dickens also acknowledges that the prior art used windowing in a different manner – this manner did not involve rewriting the stored data in a permanent database. Rather, special purpose software applied windowing and determined the four-digit year date represented by a two-digit date. This information was added to the information derived from the database, but only added in electronic memory so that the database (embodied in some semi-permanent or non-volatile memory) was never rewritten. The information in the electronic (volatile) memory, however, could be used to properly interpret the dates and execute whatever function the program required, see Dickens' mention of Shaughnessy at p. 19 – of the December 2002 reply.

Notwithstanding all of the foregoing, Dickens now, for the first time, argues his invention relates to a different application of windowing, different from the two embodiments just described.

Dickens now argues that his invention takes **all** the two-digit year dates from a permanent or non-volatile form of the legacy database and stores it elsewhere, i.e., in electronic memory. Then, applying the windowing technique in the conventional fashion, he reformats the two-digit year dates so as to properly indicate the correct century, e.g., he converts the two-digit year dates in electronic memory into four-digits. Having now converted the information from the database (but without re-writing the legacy form of the database) sorting or other operations can proceed without misinterpreting the year date information. When the particular processing has been concluded, the reformatted dates are discarded, e.g., the non-volatile or legacy form of the database is not rewritten.

The validity of Dickens' arguments can be gauged by their inconsistent application. In the main he argues that he is entitled to a patent because the art did not appreciate the advantage in **not** rewriting the legacy database, see p. 20-21 and 25-26 of the December 2002 reply (even though he admits the teaching of this feature must be implied since it is not expressly described). However it is instructive to see Dickens argue that claim 9 is patentable over Shaughnessy because Shaughnessy **does not teach rewriting** the database! (see pp 39-40 of the December 2002 reply).

THE PATENT CLAIMS

This new argument of Dickens has **no application to the patent** claims. Not one claim in the patent is directed at this subject matter.

Claim 1, for example, has four steps. The providing, selecting and determining steps are not involved in Dickens' new argument. Rather, the reformatting step, Dickens alleges, somehow incorporates the foregoing distinction. Requestor asserts, that there is nothing in this claim that corresponds to Dickens' new argument because the claim does not specify where the reformatting occurs; indeed, the claim does not even specify that there is a non-volatile form of database so as to distinguish between the information in the non-volatile form of the database from information in electronic memory. Note that the database is defined to include "symbolic representations of dates" (see the "providing" step). The reformatting step expressly operates on "**the** symbolic representations of the date". There is no hint in this claim that there might be a legacy (permanent or semi-permanent database or non-volatile database) as well as a copy of the

information from the permanent database which is found in an electronic (or volatile) memory. None of the dependent claims 2-8 has language which bears on Dickens' new argument. Claim 9, however, **contradicts** Dickens' new argument. Claim 9 depends from claim 1 and specifies "an additional step, after the step of reformatting, of storing the symbolic representation of dates and their associated information back into the database." If, as now argued, the database is not re-written, just where is this "storing" step accomplished. Conventionally, data read operations are non-destructive. There is no reason to store data back into the database unless that data was changed. Changing the database, after the reformatting is just the operation that Dickens now argues occurs only in the prior art and not in his invention. If that is the case, how to explain claim 9, a claim which appears in the Dickens application as filed!

The only other independent claim in the original 15 claims in the patent is claim 11. Claim 11 includes four similar steps. Claim 11 differs from claim 1 in that the format of the information at the conclusion of the reformatting is specified, e.g., it has the form C_1C_2 Y_1Y_2 M_1M_2 D_1D_2. This differs from claim 1 because claim 1 does not specify the format of the reformatted data, it merely identifies the information that is contained in the reformatted data. That is, claim 1 specifies that the reformatted data has values C_1C_2, and Y_1Y_2, and M_1M_2, and D_1D_2 but it does not indicate the format of that data.

However, from the point of view of Dickens' new argument, there is nothing in either claim 1 or claim 11 to specify that there is a non-volatile or legacy (permanent) form of database, that the information is read from the non-volatile form of the database and loaded into the volatile form (electronic memory) of the database and that the reformatting occurs in the electronic memory. None of claims 12-13 and 15 has language which bears on Dickens' new argument. Claim 14, however, depends from claim 11 and recites " including an additional step, after the step of sorting, of storing the sorted dates and their associated information back into the database." In other words claim 14, like claim 9 **contradicts** Dickens' new argument

It follows, that since this distinction is not contained in the claims 1-15, the claims cannot be considered patentable on this basis.

Actually, there is another and more important reason why the claims could not be entitled to distinguish from the art based on Dickens' new argument – and that is for the reason that the

subject matter of the new argument is not disclosed anywhere in the specification or drawings (either of the patent as issued, or the application as filed) and, in fact, both the patent text as issued and the application (specification and drawings) as filed are **contrary** to the substance of Dickens' new argument.

The Reissue Claims

The reissue application includes new claims 16-76, including independent claims 16, 26, 31-34, and 60-76.

Claims 16 and 26 also are four-step method claims, with steps of providing, selecting, determining and reformatting, similar to the steps of claims 1 and 11. The providing, selecting and determining steps of these claims cannot depend on Dickens' new argument for reasons similar to those advanced with respect to the patent claims.

The reformatting steps of claims 16 and 26 are similar to the reformatting steps of claims 1 and 11 except that they have added the indication that the reformatting occurs "without the addition of any new data field" and the reformatting is effected "to facilitate collectively further processing …". No part of the reformatting step specifies that information from a non-volatile or permanent database is stored in electronic memory where the information is reformatted. Accordingly, the reformatting steps of claims 16 and 26 do not incorporate Dickens' new argument. Claims 17-25 depend from claim 16. Claims 17-19 and 21-23 cannot depend on Dickens' new argument. Claim 20 recites that the reformatting occurs "separately from the symbolic representations in the database". To understand what this means a review of the claim is in order. Claim 16 recites:

> providing a database with symbolic representations of dates stored therein ...;
>
> .....
>
> reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database.

Claim 20 recites:

The method of claim 16, wherein the step of reformatting includes the step of reformatting each symbolic representation of a date into the format $C_1C_2Y_1Y_2M_1M_2D_1D_2$ separately from the symbolic representations in the database.

Claim 20 appears to contradict the parent claim 16. Claim 16 says that the database has "symbolic representations of dates stored therein". Claim 20 then says that the reformatting, which claim 16 specifies is "reformatting **the** symbolic representation" (emphasis added) but it occurs " separately" from the database. This is hardly possible since the symbolic representations which are reformatted are the representations in the database, as specified in parent claim 16. Claims 20 and 21 (which depends on claim 20) are unpatentable for lack of the required written description.

Claims 24 depends from claim 16 and recites:

The method of claim 16, including an additional step, after the step of reformatting, of storing the symbolic representation of dates and their associated information back into the database.

Claim 29 depends from claim 26 and recites:
The method of claim 26, including an additional step after the step of sorting, of storing the sorted dates and their associated information back into the database.

Both claims 24 and 29 contradict Dickens' new argument in the same manner as claims 9 and 14 contradict the same argument.

Claim 31 is also a four-step method claim but again, the providing, selecting and determining steps cannot depend on Dickens' new argument. The reformatting step differs from the reformatting step of claims 1 and 11 by adding that the reformatting occurs "without the addition of any new data field to the database" and indicates that the reformatting is "in order to facilitate collectively". Thus, for the reasons already mentioned, claim 31, like claims 1, 11, 16 and 26, does not incorporate the new distinction argued by Dickens.

Of the other independent claims, 32-34 and 60-76, all but claims 33, 34, 60-65, 70 and 71, also fail to incorporate the alleged distinction in Dickens' argument.

In connection with claims 32, 66-69 and 72-76, the reformatting clause of claim 32 calls for:

> "Reformatting the symbolic representation of each of the dates in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C_1C_2, Y_1Y_2, in order to facilitate collectively …"

The reformatting clause of claim 66 calls for:

> "reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in a database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C_1C_2, Y_1Y_2, M_1M_2, and D_1D_2".

The reformatting clause of claim 67 calls for:

> "reformatting the symbolic representation of each symbolic representation of a date in a portion of the at least one date field in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C_1C_2, Y_1Y_2."

The reformatting clause of claim 68 calls for:

> "reformatting the symbolic representation of each symbolic representation of a date in at least one date field in the database, without the addition of any new data field to a database, with the reformatted symbolic representation of each date in the database having the values C_1C_2, Y_1Y_2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates, …".

The reformatting clause of claim 69 calls for:

> "reformatting the symbolic representation of each symbolic representation of a date in the at least one date field in a database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C_1C_2, Y_1Y_2".

The reformatting clause of claim 72 calls for:

"reformatting the symbolic representation of each symbolic representation of a date in the database with the values C_1C_2, Y_1Y_2, M_1M_2, and D_1D_2 prior to collectively further processing information can contained within a database associated with the respective dates."

The reformatting clause of claim 73 calls for

"reformatting the symbolic representation of the date with the values C_1C_2, Y_1Y_2, to facilitate further processing of the dates.

The reformatting clause of claim 74 calls for "reformatting each date in the form C_1C_2 Y_1Y_2 to facilitate further processing of the dates".

The reformatting clause of claim 75 calls for:

"reformatting the symbolic representation of each symbolic representation of a date in the database, without the addition of any new data field to the database, with the reformatted symbolic representation of each date in the database having the values C_1C_2, Y_1Y_2, M_1M_2, and D_1D_2, in order to facilitate further processing of the reformatted symbolic representations of each of the symbolic representations of each of the dates."

The reformatting clause of claim 76 calls for:

"reformatting the symbolic representation of each symbolic representation of a date in the database without the addition of any new data field to the database with the reformatted symbolic representations of each date in the database having the values C_1C_2, Y_1Y_2, M_1M_2, and D_1D_2 in order to facilitate further processing of the reformatted symbolic representation of each of the symbolic representations of each of the dates".

None of these clauses specifies the presence of a permanent or non-volatile memory which remains unchanged after a reformatting or conversion in electronic or volatile memory. Accordingly none of these claims depends on Dickens' new argument.

Each of claims 16-32, 66-69 and 75-76 call for "reformatting" without the addition of any new data fields". The Shaughnessy patent and the two Japanese patent publications add the century designation without the addition of any new data fields - thus these references preform the reformatting "without the addition of any new data fields", just as claimed in claims 16-32, 66-69 and 75-76.

Dealing now with the claims 33, 34, 60-65, 70 and 71, the reformatting clause of claim 33 calls for:

> "reformatting the symbolic representation of each symbolic representation of a date in the database, without changing any of the symbolic representations of a date in the database during the reformatting step"

This language does appear to depend on Dickens' new argument. However for the reasons presented in the discussion of claim 33 in the Construction of the Claims this clause of claim 33 lacks description support.

The running a program clause of claims 34-65, 70 and 71 call for sorting or manipulation of "the converted symbolic representations" to be separate from the "representations ... in the ... database." This language also appears to depend on Dickens' new argument. Because the quoted phrase lacks description support, that argument is unavailing. However, for reasons which are set forth in connection with these claims, under the heading Construction of the Claims, an imaginative "construction" of the claims may save them from lacking any description support - however, this same reading also fails to save the claims from reading directly on references cited herein.

All three of the "sorting separate" (claims 34, 60-65, 70 and 71), "storing separate" (claims 62-65 and 71) and the "without modifying" (60-61, 64-65 and 70) clauses appear to depend on Dickens' new argument. In respect of the "sorting separate" clause (34, 60-65, 70 and 71) there is a serious question of lack of description as noted under the heading Construction of the Claims. If the claims can be reconciled with the limited description in the specification, then for the reasons given earlier, the references are not distinguished on this basis. In respect of the

"storing separate" from the database clause (62-65 and 71) there is simply no written description for the reasons specified under the heading Construction of the Claims. In respect of the converting "without modifying" clause (60-61, 64-65 and 70) the clause is without written description for the reasons specified under the heading Construction of the Claims. By ignoring aspects of the Dickens specification this clause might be shoehorned into the limited specification. In this case, however, the clause will fail to distinguish from the references.

RESPONSE TO DICKENS' NEW ARGUMENTS

Dickens' response to the rejection is composed of arguments which:

1. attempt to distinguish references on subject matter which is not found in the claims;
2. are based on incorrect statements of the law or practice;
3. include inaccurate quotations from the patent or the references; and
4. attribute substance to Exhibit A or the Certificate of Correction which are simply not supported by those documents.

At p. 21 Dickens purports to quote the specification, but the quote is foreshortened and inaccurate. Dickens' quote refers to "CCYY..." when the correct quote is "CC,YY,MM and DD". On the same page Dickens attempts to convince the reader that the invention provides for date conversion "without changing the underlying data fields of the legacy database from which the date data information was originally obtained" and that to perform the method of the Dickens patent "the converted dates **must** be stored somewhere outside of the existing database date data fields". (emphasis added) This reader however failed to glean this information from the specification. It seems that Dickens would like to forget that the specification provides:

> FIG. 1 schematically depicts a computer 20 having a read-only or random-access memory 22, a mass-storage device 23, and a central processing unit 24 therein. Stored **in the memory 22 or on the mass-storage device 23** is a database 26. (column 2, lines 44-48, emphasis added)

In other words the specification expressly notes that the database may be stored in "random-access memory" so that it is not correct that the "the converted dates **must** be stored somewhere outside of the existing database date data fields". This argument is contrary to

a. fig. 2 of the application as filed (see Exhibit 5) which notes "Reformat Dates in Database",

b. claims 9, 14, 24 and 29 which recite "an additional step, after the step of reformatting, of **storing the symbolic representation of dates and their associated information back into the database**" (emphasis added),

c. original claim 1 which provided "**reformatting** the symbolic representation of **the date in the database** with the values C_1 C_2, Y_1 Y_2, M_1 M_2 and D_1 D_2." (emphasis added)

Further it is hard to reconcile the insistence that the invention does not change the legacy database with the express statements that the "symbolic date representations are converted to a more useful form" by operation of the invention. How can the "conversion" be effected "without changing"?

At p. 36 of the December 2002 reply, in connection with claim 1 Dickens asserts that "reformatting the symbolic representation of **the** date" (emphasis added) must be read to mean "all" dates. However, since the Saka patent does expressly disclose converting all dates, even if this distinction meant something relative to Shaughnessy, it has no value relative to Saka.

Dickens has also supplied a declaration of a purported "expert". The only provision allowing such a declaration (37 CFR § 1.132) provides for supplying "evidence". If there is any "evidence" in the declaration it is hidden within erroneous quotations from the patent, inaccurate characterizations of the references, conclusory allegations which fail to take account of contrary evidence and reliance on the prosecution history for written description support whereas it is only the original application which can be relied on for written description

The Declaration of the "Expert" quotes from column 1, line 57 to column 2, line 3 (paragraph 11) that "*Each date in the database is formatted* with the values C_1C_2 Y_1Y_2 ..." (see p. 4 of the Declaration). That quote is both wrong and misleading. In the first place the quote is clearly attempting to fool the reader into believing that there is some description of working with the data set CCYY rather than with CCYYMMDD which is actually disclosed. The quote is also wrong, even within its limited extent - the data set is correctly represented as "C_1C_2 ,Y_1Y_2 ..."

Another quote which is both wrong and misleading in the very same way is found several lines down on the same page 4 of the Declaration (paragraph 12).

On the very next page (p. 5) the "Expert" makes another error in quoting from claims 1 and 11. The Declaration alleges that claims 1 and 11 both refer to the dataset "C_1C_2,Y_1Y_2,M_1M_2,D_1 D_2 ". In point of fact claim 1 refers to "C_1C_2, Y_1Y_2, M_1M_2, **and** D_1D_2 " while claim 11 refers to the different dataset "$C_1C_2Y_1Y_2M_1M_2D_1D_2$ ".

On a more substantive note the "Expert" in paragraphs 37-40 (p. 13) alleges that Hazama operates on only one or two records and not all of the records. To the contrary in paragraph 0025 the Hazama patent notes that by "replacing code", i.e., reformatting, occurs with "the effect ...that year order will be guaranteed without changing the data file record length or block length ...". It is only those with a closed mind who cannot see that guaranteeing "year order" means ordering ALL of the years not just two. Nevertheless, Saka is even more explicit on this point, clearly operating on the entire file!

Most egregious is the statements in paragraph 125 (p. 34-35). The "Expert" tells us that one skilled in the art would have understood the substance of Dickens' new argument from the disclosure referenced in paragraphs 10 and 11. Paragraph 11 is the very paragraph with the wrong and misleading quotation which is referenced above! In particular he says that it would have been understood that the reformatting be accomplished "without requiring additional **or modified** date data fields" (paragraph 125, emphasis added). Certainly Dickens discloses that reformatting occur without "additional data fields" - the argument is that there is no disclosure of achieving reformatting "without ... modified date data fields". The "Expert" continues that the disclosure "specifically says" that avoiding "such a modification" is "the very reason for the

claimed invention". Unfortunately for the Dickens' new argument the patent is specific that the goal is achieving reformatted dates without ADDING NEW DATA FIELDS, see column 1, lines 53-33. There is not one word in the patent about reformatting without MODIFYING date data fields. The "EXPERT" appears to be biased in that he reaches conclusions unsupported by any evidence.

In the very next paragraph (126) the "Expert" makes assertions about the understandings one skilled in the art would reach without once citing or relying on a single piece of evidence other than Exhibit A! A copy of Exhibit A is attached hereto as Exhibit 7. Just what it is about Exhibit 7 which provides the teachings missing from the specification and drawings of the Dickens patent is never revealed. Rather the tactic seems to be to merely mysteriously cite it again and again.

In paragraph 128 the "Expert" appears to rely on the prosecution history to supply the missing disclosure - unfortunately the "written description" cannot be expanded after the application is filed so this entire argument is unavailing.

At paragraph 135 the "Expert" notes that claim 10 was amended to correct a conflict, i.e., to insure that "the reformatted dates are **not stored back into the database**" (emphasis added). However at paragraph 83 the "expert" argues that claim 24 is patentable because the references do **not** teach "**storing** the symbolic representation of dates and their associated information **back into the database**. Which is it - stored back (claim 24) or not stored back (claim 10)?

At paragraph 162 the "Expert" attempts to argue the presence of written description for sorting on CCYY. He says, for example that this data format was known and that it would be useful and that claiming CCYY is just a broader genus. What he fails to do is to describe the written description for the subject matter under the rules laid down by the Courts - indeed he never mentions any rule or decision in the entire argument!

At paragraphs 164 and 168 the "Expert" relies on Exhibit A (attached here as Exhibit 7) and the Certificate of Correction (attached as Exhibit 8) to argue written description for the "without modifying" or "without changing" subject matter. Other than citing these documents he

never mentions just where we can look, in the document to find the support! Requestor asserts that neither document has any more description than the specification itself - namely - none.

At paragraph 177 the "Expert" admits that the terms "windowing" and "pivot year" are not found in the Dickens specification. He argues that nevertheless those skilled in the art would have understood the disclosure to contemplate the claimed subject matter, i.e., "by windowing the symbolic representations of each of the respective dates as stored in the at least one date field of the database against **a pivot year represented by one of the symbolic representations of the dates as stored in the at least one date field of the database**" (emphasis added). The Declaration is silent on whether those skilled in the art would know that the "pivot year" had to be a year "represented by one of the symbolic representations of the dates as stored in the at least one date field of the database" as claimed.

Finally, to cap off this effort the "expert" tells us (footnote 6 on p. 59) that while he has argued that the claims are patentable because one or two of the references do not teach "reformatting each" date - that the claims should not be limited to ALL - something less than ALL is actually good enough. More important than whether or not ALL is really ALL is the disclosure of Saka which does teach reformatting ALL of the dates.

NEW ISSUE OF PATENTABILITY

The references cited here raise the following new issues of patentability:

Each of claims 1-6, 8-11, 13-21, 23-26 and 28-76 is anticipated by the Saka Japanese patent publication (Exhibit 4) under 35 USC § 102 (b) as explained in the attached claim charts.

The only other claims 7, 12, 22 and 27 call for a conversion of the data in the database from a format other than YYMMDD (such as MMDDYY or YYMMMDD) to the YYMMDD format. While this is a trivial feature, well known in the art, it is not expressly described in the Saka patent publication. It is, however clearly described in the Shaughnessy patent (Exhibit 1), see in particular column 8, lines 18-27. Since Shaughnessy expressly states that the conversion can be used "regardless" of the format in the database, he also expressly motivates the

modification to the Saka disclosure. Consequently these claims are unpatentable under 35 USC § 103 (a) in view of the Saka patent publication taken with the Shaughnessy patent.

Claims 9-10, 14-15, 24-25 and 29-30 specify that the reformatted dates are stored back into the database. As shown in the claim charts these claims are read on the Saka patent publication with reliance on the key file 3 as a part of the database. Since the key file 3 is stored in electronic memory and stores reformatted dates (CCYY) and since the Dickens patent does not distinguish between a database stored in mass storage (non-volatile memory) or in electronic memory (volatile), see 2:44-48, this reading is justified.

While reliance is placed on the Saka patent publication the other cited references Shaughnessy, Hazama and Ohms are also pertinent to those of the Dickens claims which are actually supported by the skimpy disclosure.

Requestor acknowledges that the reliance on the Saka publication is different from reliance on the Hazama publication. While the differences between Saka and Hazama might appear trivial, in fact Saka expressly includes some subject matter which can only be found by implication in Hazama. Two examples include the express statement in Saka that processing (reformatting) is performed on "each record" and the indication that the database 1 has two digit year dates YY which is written to a working region of memory (random access memory) where the key file is compiled to include the century indications CC.

The Commissioner is hereby authorized to charge any deficiency in the fees filed, asserted to be filed or which should have been filed herewith (or with any paper hereafter filed in this application by this firm) to our Deposit Account No. 22-0185, under Order No. 22058-00003-US. A duplicate copy of this paper is enclosed.

Respectfully submitted,

By_____

Stanley B. Green
Registration No.: 24,351
CONNOLLY BOVE LODGE & HUTZ, LLP
1990 M Street, NW, Suite 800
Washington, DC 20036
Tel: 202-331-7111
Fax: 202-293-6229
Attorneys for Applicant

Certificate of Service

This is to certify that a copy of the attached Request for Reexamination of US Patent 5,806,063 and Appendix for the Request for Reexamination of US Patent 5,806,063 was served this 7th day of February 2003 by First Class Mail addressed to :

William C. Cray, Esq.
Goodman & Cray LLP
414 W. Fourth Street, Suite A
Santa Ana, California 92701

by_____

Stanley B. Green
Registration No. 24,351

PTO –P001832

Japanese Kokai Patent Application
No. Hei 5[1993]-27947

METHOD OF GUARANTEEING YEAR ORDER

Masakazu Hazama

UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON D.C. MARCH, 20000
TRANSLATED BY THE RALPH MCELROY TRANSLATION COMPANY

Code: PTO: 00-1832

JAPANESE PATENT OFFICE
PATENT JOURNAL (A)
KOKAI PATENT APPLICATION NO. HEI 5[1993]-27947

Int. Cl.[5]:	G 06 F 7/24 15/02
Sequence Nos. for Office Use:	8323-5B 9194-5L
Filing No.:	Hei 3[1991]-177982
Filing Date:	July 18, 1991
Publication Date:	February 5, 1993
No. of Claims:	1 (Total of 4 pages)
Examination Request:	Not filed

METHOD OF GUARANTEEING YEAR ORDER

Inventor:	Masakazu Hazama
Applicant:	000005108 Hitachi, Ltd.

[There are no amendments to this patent.]

Claim

1. Method of guaranteeing year order characterized in that, in a computer system that has a memory means and a processing section, when the last 2 digits for years in the 1900's and 2000's AD are stored in the aforementioned memory means, the processing section replaces the code for the 10's place in the last 2 digits of the year AD with a code that maintains the year order.

Detailed explanation of the invention

[0001]

Industrial application field

This invention pertains to a method of guaranteeing year order for the year 2000 AD that guarantees ascending/descending order with evaluation of magnitude and sort/merge processing using programs for digital files in which the last 2 digits of the year AD are stored, and that can obtain correct results.

[0002]

Prior art

With conventional computer systems, the last 2 digits of years AD are stored. Note that, for example, Japanese Kokai Patent Application Nos. Sho 58[1983]-1229 and Hei 03[1991]-22117 involve this type of technology.

[0003]

Problems to be solved by the invention

The aforementioned prior art does not take into consideration years after 2000 AD, and manages the year with the last 2 digits of the year AD in a file. For this reason, after 2000 AD, when ascending/descending order is handled by processing that evaluates magnitude and by sort/merge processing using normally numbered years, their relative magnitudes are represented by Formula 1.

[0004]

Formula 1

$$1999 > 1998 > 2001 > 2000$$

[0005]

That is, regardless of the fact that the year 2000 must be evaluated to be larger than the year 1999, for evaluation, only the last 2 digits of the date are used, so since 00 is smaller than 99, year 2000 is evaluated to be smaller than year 1999. Using 4 digits for the date has been considered as a method of resolving this, but in this case, it is necessary to change the data file record length and block length, and program modifications also arise.

[0006]

The purpose of this invention is to provide a method of guaranteeing year order for handling 2000 AD that makes use of the code system and that can also handle years after 2000 AD.

[0007]

Means to solve the problems

To accomplish the aforementioned purpose, in a computer system that has a memory means and a processing section, when the last 2 digits for years in the 1900's and in the 2000's AD are stored in the aforementioned memory means, the processing section will replace the code of the 10's place in the last 2 digits of the date with a code that maintains the year order.

[0008]

Function

When there are data present that indicate years in the 1900's and 2000's AD, the data code that represents the date is replaced by another code so that the year order will be maintained. In this way, magnitude evaluation and ascending/descending order processing are guaranteed.

[0009]

Application example

Figure 1 is a block diagram of a program in one application example of this invention.

[0010]

(6) is a file where data that include only the last 2 digits of the year AD are stored. (7) is a program. (8) is a clear area. (9) is a parameter. (10) is a 2000 AD correspondence utility module.

[0011]

2000 AD correspondence utility module (10) (hereafter called module (10)) is activated when specified by program (7) or a utility and is positioned as pre-processing for processing that handles the year. Note that, in module (10), a range of the last 2 digits for which code transformation will be performed is specified in advance. Replacement involves numbers for years in the 2000's where the last 2 digits are smaller than the smallest number in the last 2 digits in years in the 1900's. For example, when data in file (6) for years AD begin with the year 1973, the last 2 digits are replaced using 00 (year 2000) for 72 (year 2072). The present application example is an example where there are data from year 1960 in file (6), such a range is specified

so that the last 2 digits will be transformed to codes 00-59. Note that, in the present application example, EBCDIC code is used as the code.

[0012]

Next, the processing sequence will be explained. First, program (7) calls module (10) at a preliminary stage that evaluates the year. After that, the following processing is performed by module (10).

[0013]

Parameter analysis processing (1): parameter (9), provided from outside, is input and the contents are analyzed, and the starting positions of the work area name and the last 2 digits of the year AD in the record are confirmed.

[0014]

Work area input processing (2): data from the work area name obtained by parameter analysis processing (1) are input.

[0015]

Year evaluation processing (3): 2 bytes from the starting position of the last 2 digits in the year AD in the record obtained by parameter analysis processing (1) are evaluated, and if within a fixed range, in the present application example, in the range from '00' to '59,' replacement processing (4) is performed. In addition to this, the next data are input.

[0016]

Replacement processing (4): the 10's place in the last 2 digits in the year AD in the record is replaced as in Table 1.

[0017]

Table 1

	置換前 (1)	置換後 (2)
(3) 十の位が０の場合	X' F0'	X' FA'
十の位が１の場合	X' F1'	X' FB'
十の位が２の場合	X' F2'	X' FC'
十の位が３の場合	X' F3'	X' FD'
十の位が４の場合	X' F4'	X' FE'
十の位が５の場合	X' F5'	X' FF'

Key: 1 Before replacement
2 After replacement
3 When 10's place is __

[0018]

Work area output processing (5): data that have undergone replacement processing (4) are output to work area (8).

[0019]

By replacing character codes as shown in Table 1, it is evaluated that year 2000 is greater than year 1990, and that year 2010 is greater than year 2000.

[0020]

Note that, in the present application example, the replacement processing with codes shown in Table 1 was performed for [years] greater than year 2000 with EBCDIK code, but in the case of [years] less than year 2000, they could also be replaced by empty code as in Table 2. With this method, for example, if there are data from year 1999 in file (6), up to year 2098 can be handled.

[0021]

Table 2

	置換前(1)	置換後(2)
(3) 十の位が０の場合	X' F 0'	X' B 0'
十の位が１の場合	X' F 1'	X' B 1'
十の位が２の場合	X' F 2'	X' B 2'
十の位が３の場合	X' F 3'	X' B 3'
十の位が４の場合	X' F 4'	X' B 4'
十の位が５の場合	X' F 5'	X' B 5'
十の位が６の場合	X' F 6'	X' B 6'
十の位が７の場合	X' F 7'	X' B 7'
十の位が８の場合	X' F 8'	X' B 8'
十の位が９の場合	X' F 9'	X' B 9'

Key: 1 Before replacement
2 After replacement
3 When 10's place is __

[0022]

Also, for [years] less than year 2000, they could also be replaced with code that uses X'F0' → X'C0', etc.

[0023]

Also, in the case of years in the 2000's, they could be replaced by X'F0' → X'C0', and for years in the 1900's, replaced by X'F0' → X'B0'. That is, both years in the 2000's and in the 1900's could be replaced by other codes.

[0024]

Also, it makes no difference if the character code used is JIS code or ASCII code, etc. That is, when the relative magnitudes of years are evaluated, code replacement need only be performed so that evaluation is correctly accomplished.

[0025]

Effect of the invention

By replacing code so that the relative magnitudes of years are correctly evaluated, the effect is that year order will be guaranteed without changing data file record length or block length, and further, without modifying programs.

[0026]

Brief description of the figures

Figure 1 is a figure that shows program configuration.

Explanation of symbols

(1) ... parameter analysis processing, (2) ... work area input processing, (3) ... year evaluation processing, (4) ... replacement processing, (5) ... work area output processing, (6) ... file, (7) ... program, (8) ... work area, (9) ... parameter, (10) ... 2000 AD correspondence utility module.

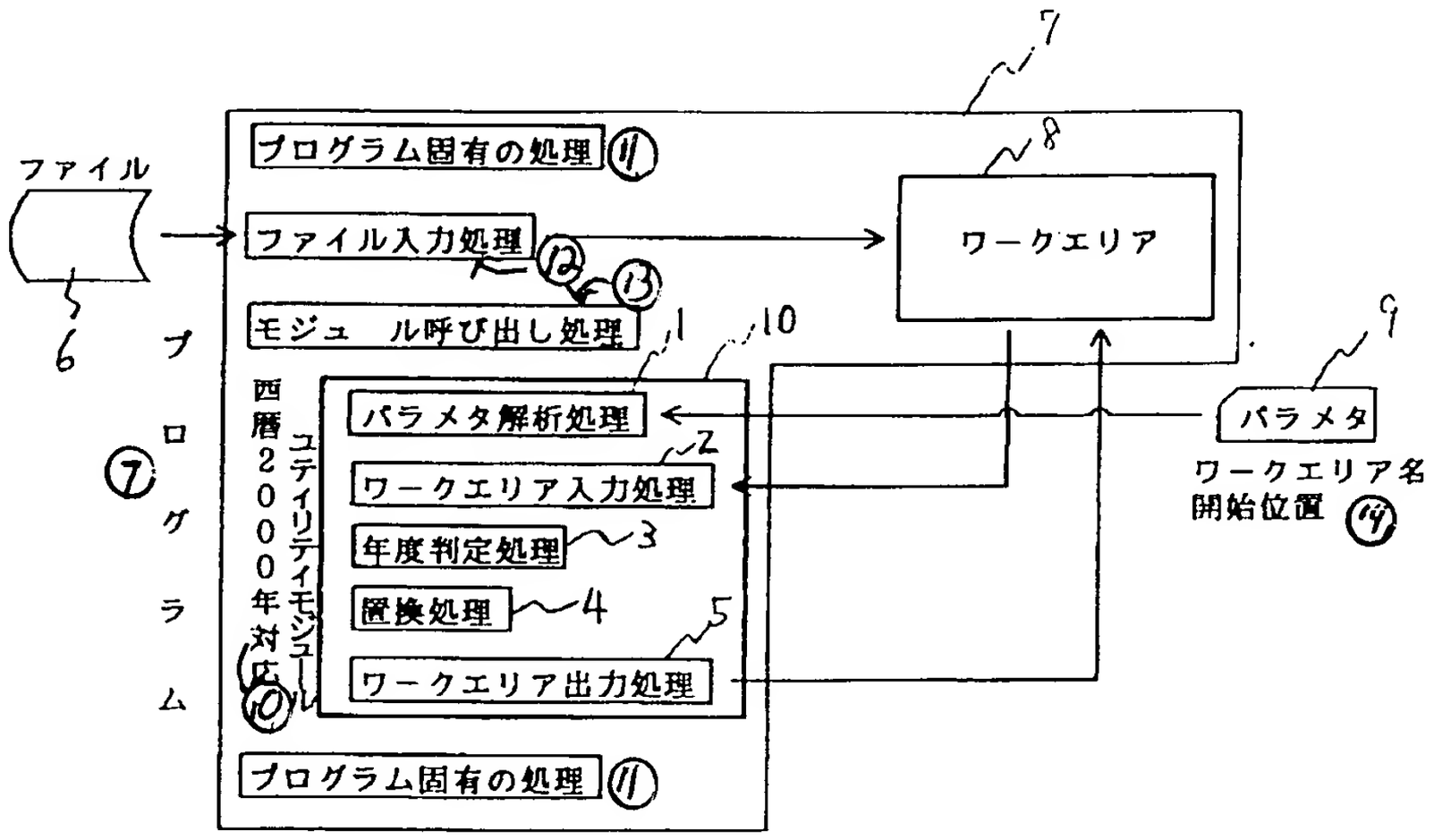

Figure 1

Key: 1 Parameter analysis processing
2 Work area input processing

3	Year evaluation processing
4	Replacement processing
5	Work area output processing
6	File
7	Program
8	Work area
9	Parameter
10	Year 2000 date correspondence utility module
11	Processing unique to program
12	File input processing
13	Module call-up processing
14	Work area name start position